UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/025,217	12/18/2001	Igor Liokumovich	42P12564	5026

59796 7590 01/14/2008

INTEL CORPORATION
c/o INTELLEVATE, LLC
P.O. BOX 52050
MINNEAPOLIS, MN 55402

EXAMINER	
GUILL, RUSSELL L	
ART UNIT	**PAPER NUMBER**
2123	

MAIL DATE	DELIVERY MODE
01/14/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/025,217	LIOKUMOVICH ET AL.	
	Examiner	Art Unit	
	Russ Guill	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 December 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-5 and 7-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-5 and 7-28* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *18 December 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is in response to an Amendment filed December 13, 2007. No claims were added or canceled. Claims 1, 3 - 5 and 7 – 28 are pending, and have been examined. Claims 1, 3 - 5 and 7 – 28 have been rejected.

2. This Office Action is NON-Final due to new rejections.

Response to Remarks

3. Regarding claims 1, 9, 16 and 23 rejected under 35 USC § 103:

3.1. Applicant's arguments have been fully considered, and are persuasive, as follows. However, upon further search and consideration, new rejections are made below.

3.2. The Applicant argues:

4. The Examiner asserts that Klaiber teaches *the monitor modifying the original values in a descriptor table to prevent the translated code from being accessed...* However, Klaiber teaches setting a bit in the page's entry in the processor's memory management unit (MMU). Klaiber then describes that this bit is invisible to X86 software. Klaiber's method modifies the actual page table and not a descriptor table.

5. In contrast, descriptor tables are known in the art, as are segment tables/descriptors. For instance, a definition for a segment descriptor may be found at

6. en*wikipedia*org/wikifSegment descriptor that reads:

7. "In memory addressing for Intel x86 computer architectures, segment descriptors are a part of the segmentation unit, used for translating a logical address to linear address. Segment descriptors describe the memory segment referred in the logical address.

8. A logical address in Intel x86 consists of a segment selector and an offset. The most significant 13 bits of the segment selector defines the address of the segment descriptor, which is stored either in the Global Descriptor Table (GDT) or Local Descriptor Table (LDT). The description details mainly include the memory segment's first byte in

linear address (Base), size (Limit) and Type (Bovet & Cesati, 2000, p. 36 - 41)."

8.1. Descriptor tables are also known in the art and described in the specification. The descriptor table with the segment table is described as being modified by the virtual machine monitor. The segment table in the descriptor table is not the same thing *as* a "page's entry in the MMU." Segmentation is one form of address translation that happens prior to paging. Page table entries are a separate mapping function.

9. Moreover, Klaiber teaches write-protecting a page of memory. The modification as claimed by Applicants prevents the translated code from being accessed, not an entire page of memory. As described by Applicant, segmentation provides a mechanism for dividing the processor's linear address space into smaller protected regions called "segments." The operating system defines its segments by assigning a segment base, a segment limit, and different segment attributes, e.g., type, granularity, DPI. (Descriptor Privilege Level). In contrast, pages are defined by the MMU. Thus, the prior art fails to show each limitation of the recited claims.

9.1.1. The Examiner respectfully replies:

9.1.2. While the Examiner disagrees with the Applicant's arguments, since segment tables are readily available in the existing art, the claims will simply be rejected using existing art.

9.2. The Applicant argues:

9.3. Further, Rosenblum seems to teach an operating system simulator. In contrast, the claimed simulator supports a full platform simulator which, as described in the specification, includes simulation of the instruction set architecture of the processor.

9.3.1. The Examiner respectfully replies:

9.3.2. The argument does not appear to have a stated purpose or conclusion, and thus, no reply appears to be needed.

9.4. The Applicant argues:

9.5. Independent claims 1, 9, 16 and 23 have been amended to more clearly recite that segment information is modified in the descriptor tables. Thus, Claims 1, 9, 16, 23 and their progeny are believed allowable.

9.5.1. The Examiner respectfully replies:

9.5.2. The independent claims are newly rejected with the existing references.

9.6. The Applicant argues:

9.7. Claims 5, 12, 19 and 26 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Devine, Klaiber, Rosenblum and further in view of *"Running multiple operating systems concurrently on an IA32 PC using virtualiattion techniques"* by Kevin Lawton (hereinafter "Lawton"). This rejection is respectfully traversed and Claims 5, 12, 19 and 26 are believed allowable as amended based on the foregoing and following discussion.

9.8. Claims 5, 12, 19 and 26 are believed allowable, at least by being dependent on allowable base claims.

9.8.1. The Examiner respectfully replies:

9.8.2. The independent claims are newly rejected with the existing references.

10. Regarding claims 7 - 8, 14 - 15, 21 - 22 and 27 - 28 rejected under 35 USC § 103:

10.1. Applicant's arguments have been fully considered, and are persuasive, as follows.

10.2. The Applicant argues:

10.3. Claims 7, 8, 14, 15, 21-22 and 27-28 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Devine, Klaiber, Rosenblum and further in view of *"Operating systems internals and design principles"* by William Stallings (hereinafter "Stallings"). This resection is respectfully traversed and Claims 7, 8, 14, 15, 21-22 and 27-28 are believed allowable as amended based on the foregoing and following discussion,

10.4. Claims 7, 8, 14, 15, 21-22 and 27-28 are believed allowable, at least by being dependent on allowable base claims.

10.4.1. The Examiner respectfully replies:

10.4.2. The independent claims are newly rejected with the existing references.

10.5. The Applicant argues:

10.6. Further, the Examiner asserts that Stallings teaches modifying the descriptor table to remove a portion of a segment that overlaps with the memory storing the translated code, or replace a segment to create a fault. However, it seems that Stallings merely teaches basic techniques for creating memory segments.

10.7. A cursory review of the teachings of Stallings do not show a. teaching or suggestion that overlapping portions are to cause the descriptor table to be modified. It seems that Stallings is merely teaching how to set up a segmentation scheme and the translation of segment addresses. Stallings seems to teach allocating and translating segments and the handling of growing or shrinking data structures. There seems to be no teaching or suggestion for modifying the descriptor tables when segments overlap translated code. In fact, no mention of translated code seems to appear at all. Thus, Stallings fails to teach or suggest the claimed limitation.

10.8. The Examiner asserts that it would be obvious to modify Devine, Klaiber and Rosenblum with Stallings to modify a segment or replace a segment with a substitute segment to cause a fault. However. Stallings teaches only that more processes may be put in memory by overlaying them. This overlay scheme is based on original code, and not translated code. There is no suggestion in Stallings that translated code, i.e., code to be simulated, is to be written into data segments that cause an overlap of segments to the translated code. In contrast, the claimed invention translates machine instructions into translated code, and this translation may cause an overlap. The overlap here is not the same as overloading memory with too many processes in memory. The problem is different, the cause is different, and the result is different. Stallings will swap memory back and forth when a fault occurs.

10.9. In contrast, Applicants' invention transfers control to between a VM and VMM so that the translated code may be simulated. Thus, there is no suggestion in Stallings to replace segments for this purpose, nor would it be obvious to one of skill in the art. Moreover, as recited in the base claims, a modification of the segment information in the descriptor table is made to prevent the translated code from being accessed. This type of modification is neither taught nor suggested by Stallings. Thus, all of the pending claims are believed allowable.

10.9.1. The Examiner respectfully replies:

10.9.2. Applicant's arguments have been fully considered, and are persuasive.

Drawings

11. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description: Figure 1, element 152 does not appear to be recited in the specification. Figure 3A, elements 20, 32, 33, 311 and 312 do not appear to be recited in the specification. Figure 3B, elements 32, 33, 39, 391 and 392 do not appear to be recited in the specification. Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Objections

12. Claim 23 recites in line 13, "executing on a host machine on the platform". The phrase may mean, "executing on the platform". The Applicant is respectfully asked to review the phrase for possible amendment.

Claim Rejections - 35 USC § 112

13. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

13.1. Claims 1, 3 - 5 and 7 – 28 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

13.1.1. Regarding independent claims 1, 9, 16 and 23, and dependent claims: Claim 1 recites, "the simulator executing the translated code that represents simulated operating system code to be executed on the virtual machine". Claims 9 and 16 recite, "the simulator executing the translated code that represents simulated operating system code to be executed on a virtual machine". Claim 23 recites, "the simulator executing the translated code that represents simulated operating system code of the first instruction set architecture to be executed on the virtual machine in the second instruction set architecture". The recited subject matter does not appear to be described in the specification.

13.1.2. First, the antecedent basis for "the simulator" appears to be "a full platform simulator that includes device models", but the full platform simulator (*paragraphs [0015] - [0016]*) does not appear to execute the translated code that represents the simulated operating system code. As recited in paragraph [0017], the simulated operating system code runs in the VM.

13.1.3. Second, it appears to be contradictory that the simulator executes translated code that will be executed on the virtual machine, and also does not appear to be supported in the specification.

13.1.4. Third, the specification does not appear to recite that the operating system code is translated. The invention appears to have a separate memory space allocated for translated code (*figure 1, element 155, and paragraph [0020]*), that is separate from the simulated operating system (*figure 1, element 151*).

Claim Rejections - 35 USC § 103

14. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

15. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

16. **Claims 1, 3, 4, 9, 10, 11, 13, 16, 17, 18, 20 and 23 - 25** are rejected under 35 U.S.C. 103(a) as being unpatentable over Devine (U.S. Patent 6,397,242, May 28, 2002) in view of Klaiber (Klaiber, Alexander; "The Technology Behind Crusoe Processors", January 2000, www.transmeta.com/pdfs/paper_aklaiber_19jan00.pdf), further in view of Rosenblum (Mendel Rosenblum et al.; "Complete Computer System Simulation: The SimOS Approach", Winter 1995, IEEE Parallel & Distributed Technology, pages 34 - 43).

16.1. Regarding claims 1, 9, 16 and 23:

16.1.1. Devine appears to teach a monitor executing on the host machine that translates the machine instructions into translated code (***column 8, lines 1 - 4; column 21, lines 60 - 67; and column 22, lines 1 - 21***).

16.1.2. Devine appears to teach the monitor modifying original values of segment information in a descriptor table (***figure 5 shows segment information in descriptor tables, including protection bits, and column 15, lines 24 - 67, column 16, lines 1 - 32, especially note the protection bits; further, please note that protection bits in a segment table were common knowledge in the art, see Stallings, page 324, figure 8.2(b), and page 325, second paragraph***);

16.1.3. Devine appears to teach a virtual machine executing on the host machine that executes the translated code stored in memory (***column 10, lines 51 - 54; and column 22, lines 7 - 8; and column 25, lines 23 - 46 [especially lines 36 - 46]***).

16.1.4. Devine appears to teach a kernel executing on the host machine that detects exceptions occurring in the virtual machine and transfers control between the virtual machine and the monitor according to the type of exception (***column 14, lines 56 - 62; and column 17, lines 33 - 39; and column 17, lines 44 - 50; and column 22, lines 45 - 55***).

16.1.5. Devine does not specifically teach:

16.1.5.1. ~~a monitor executing on the host machine that translates the machine instructions into translated code, the monitor modifying original values of segment information in a descriptor table~~ ***to prevent the translated code from being accessed, thereby preventing the translated code from being modified***.

16.1.5.2. wherein an operating system executing on the host machine also supports a full platform simulator that includes simulation modules and device models, the simulator executing the translated code that represents simulated operating system code to be executed on the virtual machine, and wherein results of executing the translated code are provided to a user.

16.1.6. Klaiber appears to teach the monitor modifying original values ~~of segment information in a descriptor~~ table to prevent the translated code from being accessed, thereby preventing the translated code from being modified (***Klaiber, page 14, section "Coping with Self-modifying code", fourth sentence***).

16.1.7. Rosenblum appears to teach:

16.1.7.1. an operating system executing on the host machine also supports a full platform simulator that includes simulation modules and device models (***page 35, figure 1, "SimOS target hardware layer" and "Host Platform" layer***), the simulator executing the translated

code that represents simulated operating system code to be executed on the virtual machine (*page 35, right-side column, section "The SimOS environment", , and page 36, left-side column, first paragraph, and fourth paragraph; and page 36, right-side column, second paragraph that starts with, "SimOS also includes . . "*), and wherein results of executing the translated code are provided to a user (*page 35, left-side column, third paragraph that starts with, "The other feature . . .", sentence that starts with, "Statistics collected "; it would have been obvious that the statistics were provided to a user*).

16.1.8. The motivation to use the art of Rosenblum with the art of Devine would have been the multiple benefits recited in Rosenblum, including an extremely fast simulation of system hardware (*page 35, left-side column, second paragraph*) and a fast simulation mode that allows positioning of long-running workloads is essential for performance studies (*page 43, left-side column, second paragraph*); which would have been recognized as benefits by the ordinary artisan.

16.1.9. The motivation to use the art of Klaiber with the art of Devine is the benefit recited in Klaiber of a solution that combines strong performance with remarkably low power consumption (*Klaiber, page 3, section "Summary", first sentence*). Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Klaiber and the art of Rosenblum with the art of Devine to produce the claimed inventions.

16.2. Regarding claims 3, 11, 18 and 24:

16.2.1. Devine appears to teach that the translated code and the original machine instructions access the memory using a same set of addresses as a set of addresses used by the original machine instructions (*column 10, lines 50 – 55; and column 1, lines 45 - 67*).

16.2.1.1. Regarding (*column 10, lines 50 – 55; and column 1, lines 45 - 67*); since the virtual machine directly executes instructions on the underlying hardware, it is obvious that the translated code and the original machine instructions access the memory using a same set of addresses as a set of addresses used by the original machine instructions.

16.3. Regarding claims 4 and 25:

16.3.1. Devine appears to teach that the monitor further includes an auxiliary simulator that executes the machine instructions (*column 21, lines 61 – 67; and column 22, lines 1 – 6*).

16.3.1.1. Regarding (*column 21, lines 61 – 67; and column 22, lines 1 – 6*); since the translator calls the VMM to execute certain machine instructions, it is obvious that the monitor further includes an auxiliary simulator that executes the machine instructions.

16.4. Regarding claims 13 and 20:

16.4.1. Devine appears to teach ~~the monitor modifies the descriptor table to prevent the translated code from being modified~~, the descriptor table including attributes of a segment of the memory (*figure 5 shows segment information in descriptor tables, including protection bits, and column 15, lines 24 – 67, column 16, lines 1 – 32, especially note the protection bits; further, please note that protection bits in a segment table were common knowledge in the art, see Stallings, page 324, figure 8.2(b), and page 325, second paragraph*);

16.4.2. Devine does not specifically teach that the monitor modifies the descriptor table *to prevent the translated code from being modified*.

16.4.3. Klaiber appears to teach that the monitor modifies a table to prevent the translated code from being modified (*Klaiber, page 14, section "Coping with Self-modifying code", fourth sentence*).

16.5. Regarding claims 10 and 17:

16.5.1. Devine does not specifically teach simulating a device.

16.5.2. Rosenblum appears to teach simulating a device (*page 38, right-side column, section "Device simulation"*).

17. Claims 5, 12, 19 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Devine and Klaiber and Rosenblum as applied to **claims 1, 3, 4, 9, 10, 11, 13, 16, 17, 18, 20 and 23 – 25** above, further in view of Lawton (Lawton, Kevin; "Running multiple operating systems concurrently on an IA32 PC using virtualization techniques", www.anticracking.sk/EliCZ/import/Vx86.txt).

17.1. Devine as modified by Klaiber and Rosenblum teach a system, method and computer program product for simulating machine instructions on a host machine as described in **claims 1, 3, 4, 9, 10, 11, 13, 16, 17, 18, 20 and 23 – 25** above.

17.2. Regarding claims 5, 12, 19 and 26:

17.3. Devine appears to teach:

17.3.1. the capsule being one of a simple capsule and a complex capsule, and wherein simple capsule is executed by the virtual machine and a complex capsule is executed by the virtual machine monitor (***column 1, lines 59 - 67, and column 2, lines 1 - 5***).

17.4. Devine does not specifically teach that the monitor replaces one of the machine instructions with a capsule if the machine instruction accesses a system state of a central processing unit of the host machine.

17.5. Lawton appears to teach that the monitor replaces one of the machine instructions with a capsule if the machine instruction accesses a system state of a central processing unit of the host machine (***page 3, section "CHALLENGE ON THE IA32", paragraphs 1 - 3***).

17.5.1. Regarding (***page 3, section "CHALLENGE ON THE IA32", paragraphs 1 - 3***); trapping out is equivalent to a capsule.

17.6. The motivation to use the art of Lawton with the art of Devine as modified by Klaiber and Rosenblum would have been the benefit recited in Lawton to run a primary PC operating system and related software while retaining the ability to concurrently run software engineered for a different PC operating system (***page 1, section "THE RATIONALE FOR VIRTUALIZATION", first paragraph***). Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Lawton with the art of Devine as modified by Klaiber and Rosenblum to produce the claimed inventions.

18. **Claims 7, 8, 14, 15, 21 - 22 and 27 - 28** are rejected under 35 U.S.C. 103(a) as being unpatentable over Devine as modified by Klaiber and Rosenblum as applied to **claims 1, 3, 4, 9, 10, 11, 13, 16, 17, 18, 20 and 23 - 25** above, further in view of Stallings (Stallings, William; "Operating systems: internals and design principles", 1998, Prentice-Hall).

18.1. Devine as modified by Klaiber and Rosenblum teach a system, method and computer program product for simulating machine instructions on a host machine as described in **claims 1, 3, 4, 9, 10, 11, 13, 16, 17, 18, 20 and 23 - 25** above.

18.2. **Regarding claims 7, 14, 21 and 27:**

18.2.1. Devine does not specifically teach that the monitor modifies the descriptor table to remove a portion of a segment that overlaps with the memory storing the translated code.

18.2.2. Stallings appears to teach that the monitor modifies the descriptor table to remove a portion of a segment that overlaps with the memory storing the translated code (***page 334, section labeled "Segmentation", sub-section labeled "Virtual Memory Implications", second paragraph, especially item number 1, sentence 3; and pages 307 – 309 section 7.4 Segmentation***).

18.2.2.1. Regarding (***page 334, section labeled "Segmentation", sub-section labeled "Virtual Memory Implications", second paragraph, especially item number 1, sentence 3; and pages 307 – 309 section 7.4 Segmentation***); since the advantage of segmentation was that the operating system will shrink a segment as needed, it would have been obvious that the monitor modifies the descriptor table to remove a portion of the segment that overlaps with the memory storing the translated code.

18.2.3. The motivation to use the art of Stallings with the art of Devine is the advantage recited in Stallings that segmentation simplifies the handling of growing data structures, and the operating system will expand or shrink the segment as needed (***page 334, section labeled "Segmentation", sub-section labeled "Virtual Memory Implications", second paragraph, especially item number 1, sentence 3***). Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Stallings with the art of Devine as modified by Klaiber and Rosenblum to produce the claimed inventions.

18.3. Regarding claim 8, 15, 22 and 28:

18.3.1. Devine does not specifically teach that the monitor modifies the descriptor table to replace a segment with a substitute segment, which, when accessed, causes an exception to be generated.

18.3.2. Stallings appears to teach that the monitor modifies the descriptor table to replace a segment with a substitute segment, which, when accessed, causes an exception to be generated (***page 335, section labeled "Organization", first paragraph, especially the sentence that starts with "Because only some of the segments of a process may be in main memory . . . "; and page 324, figure 8.2b; and pages 324 - 325, section labeled "Paging"; and pages 319 – 320, section 8.1 "Hardware and Control Structures", especially page 320, the paragraph that starts with "Let us consider . . ."***).

18.3.3. The motivation to use the art of Stallings with the art of Devine is the benefit recited in Stallings that it is not necessary for all of the segments to be in memory during execution (***page 319, section 8.1 "Hardware and Control Structures"***).

19. **Examiner's Note:** Examiner has cited particular columns and line numbers in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the Applicant in preparing responses, to fully consider the references in their entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner.

Allowable Subject Matter

20. Any indication of allowability is withheld pending resolution of the outstanding rejections.

Conclusion

21. The prior art not relied upon teaches knowledge of the ordinary artisan:

    21.1. John Scott Robin et al., "Analysis of the Intel Pentium's Ability to Support a Secure Virtual Machine Monitor", August 2000, Proceedings of the 9th USENIX Security Symposium, 16 unnumbered pages; teaches common knowledge regarding virtual machine monitors.

22. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Russ Guill whose telephone number is 571-272-7955. The examiner can normally be reached on Monday - Friday 10:00 AM - 6:30 PM.

23. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul Rodriguez can be reached on 571-272-3753. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Any inquiry of a general nature or relating to the status of this application should be directed to the TC2100 Group Receptionist: 571-272-2100.

24. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Russ Guill
Examiner
Art Unit 2123

RG

PAUL RODRIGUEZ
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100